AIAA SciTech Forum
7-11 January 2019, San Diego, California
AIAA Scitech 2019 Forum

10.2514/6.2019-0690

Vehicle to Vehicle (V2V) communication for Collision avoidance for Multi-copters flying in UTM -TCL4

Anjan Chakrabarty[*], Corey Ippolito[†], Joshua Baculi[‡], Kalmanje Krishnakumar[§]

& Sebastian Hening[¶]

NASA Ames Research Center, Moffett Field, CA 94035

NASA's UAS Traffic management (UTM) research initiative is aimed at identifying requirements for safe autonomous operations of UAS operating in dense urban environments. For complete autonomous operations vehicle to vehicle (V2V) communications has been identified as an essential tool. In this paper we simulate a complete urban operations in an high fidelity simulation environment. We design a V2V communication protocol and all the vehicles participating communicate over this system. We show how V2V communication can be used for finding feasible, collision-free paths for multi agent systems. Different collision avoidance schemes are explored and an end to end simulation study shows the use of V2V communication for UTM TCL4 deployment.

Keywords: UTM, Autonomy, Path planning, BVLS, V2V communication. DSRC communication

I. Introduction

The exponential growth of Unmanned Aerial Vehicles (UAVs) for different applications like package delivery,[1] inspection,[2] security[3] and disaster management[4] has opened different commercial business opportunities. It is predicted that by 2021, 3 to 6 million air vehicles will fly in the national airspace[5] with daily operations potentially reaching millions.[6] New vehicle technologies and infrastructural capabilities are needed to effectively handle large scale operations and effectively manage the airspace.

For effective use of UAS for different applications, it is imperative that the vehicles are designed to be able to work autonomously in a co-operative manner. The industry is witnessing a rapid rise in Unmanned aircraft systems (UAS) technology to address these issues. UAS are being designed to operate autonomously and beyond visual line of sight (BVLOS) with minimal supervision. This brings new challenges in managing large fleets of vehicles. UASs also share the same airspace as large aircrafts and thus autonomous UAS operations and current airspace management has to be integrated. Safety, security, efficiency and equity of the National Airspace System (NAS) have to be handled in a comprehensive manner.

NASA is working on developing an UAS Traffic Management System (UTM) from 2013[7] to handle large scale UAS operations in the NAS. UTM is comprised of all infrastructure, policies and procedures to support low altitude UAS operations.[8] It is a community based traffic management solution where "Operators" are responsible for managing their own operations and provide situational awareness to the entire system. UTM will provide management service and co-ordination with Federal Aviation Authority (FAA).

NASA is conducting a series of endeavors named Technology Capability Levels(TCL) which is bringing UAS flight with FAA approval. Starting from sparsely populated rural areas, UTM is gradually conducting UAS flights in urban populated regions with each increasing TCL level with increasing complexity. The current progress made under different national air campaigns with different technological partners is described in.[9] It describes the series of flight tests concluded recently under TCL-3 in sparsely populated urban regions.

Concepts of TCL-4, which is operation of UAS in densely populated urban regions are still under development. It is assumed that TCL-4 will have BVLOS flight with autonomous vehicle-to-vehicle or internet connectivity.[8] A

[*] Research Engineer, SGT Inc, NASA Ames Research Center, Moffet Field, CA 94035. AIAA Member. anjan.chakrabarty@nasa.gov

[†] Aerospace Scientist, NASA Ames Research Center, Moffet Field, CA 94035, AIAA Member

[‡] Systems Engineer, HX5, LLC., NASA Ames Research Center, Moffett Field, CA 94035, AIAA Member

[§] Aerospace Scientist, NASA Ames Research Center, Moffet Field, CA 94035, AIAA Member

[¶] former Research Engineer, SGT Inc, NASA Ames Research Center, Moffet Field, CA 94035. AIAA Member.

notional UTM architecture is presented in the next section which outlines the communication between the different actors participating in UTM.

A. UTM architecture

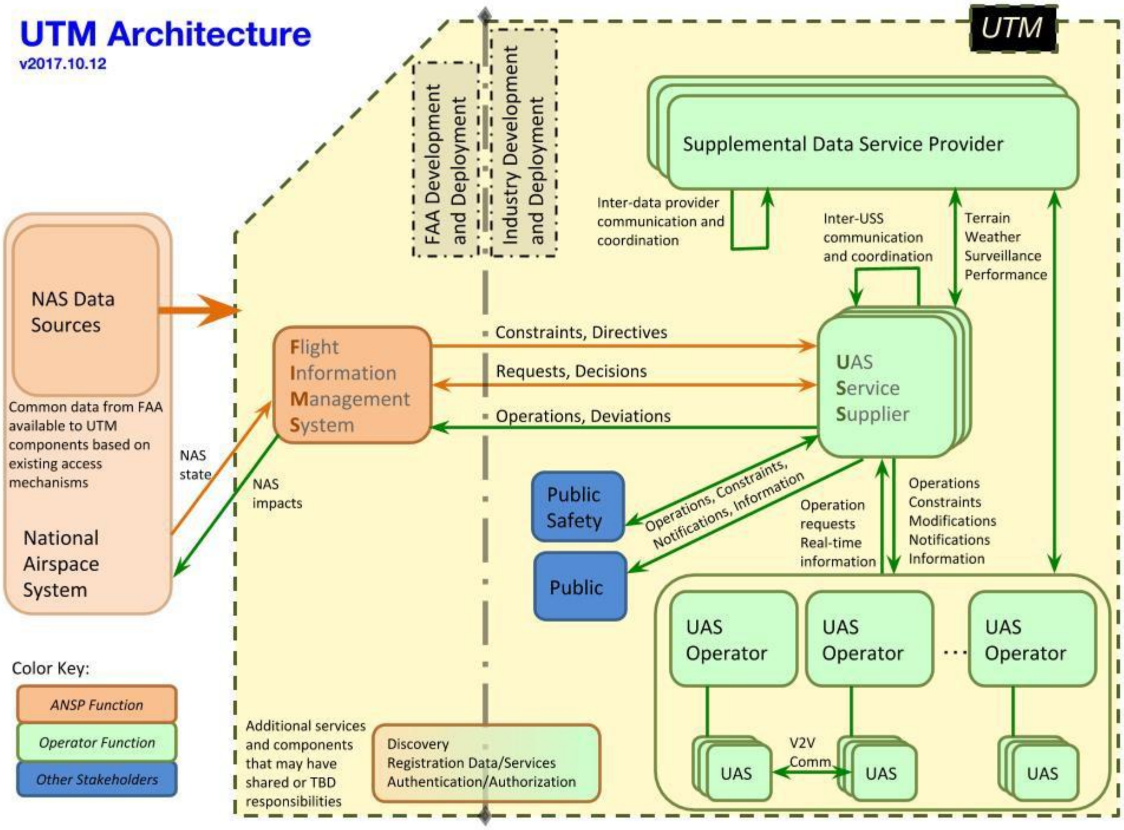

UTM Architecture
v2017.10.12
UTM
FAA Development and Deployment
Industry Development and Deployment
Supplemental Data Service Provider
Inter-data provider communication and coordination
Inter-USS communication and coordination
Terrain
Weather
Surveillance
Performance
NAS Data Sources
Common data from FAA available to UTM components based on existing access mechanisms
National Airspace System
NAS state
NAS impacts
Flight Information Management System
Constraints, Directives
Requests, Decisions
Operations, Deviations
UAS Service Supplier
Public Safety
Public
Operations, Constraints, Notifications, Information
Operation requests
Real-time information
Operations
Constraints
Modifications
Notifications
Information
UAS Operator
UAS Operator
UAS Operator
UAS
UAS
UAS
V2V Comm.
Color Key:
ANSP Function
Operator Function
Other Stakeholders
Additional services and components that may have shared or TBD responsibilities
Discovery
Registration Data/Services
Authentication/Authorization

Figure 1. Proposed UTM Architecture for TCL 4.

The Federal Aviation Administration (FAA) has outlined a UTM concepts of operations.[10] The document is based on evolution of NASA's UTM through different use case development. The FAA expects that the industry will develop solutions to assist in the management of the increasing number of UAS operations. See[10] for the full overview and use cases as envision by NASA/FAA. Here we summarize the relevant portions that are needed for this paper.

Figure 1 presents a proposed UTM architecture, that identifies the various actors and components. The UTM is designed as a decentralized communication sharing and exchanging platform. Most UAS operations are envisioned as individual operators operating their own UAS. FAA does not need to intervene and regulate each UAS operation. Rather FAA will manage only the airspace and the actors operate co-operatively in the airspace sharing situational awareness.

Flight Information Management System(FIMS) is the gateway for data exchange between the different actors. FIMS is being developed at NASA, in collaboration with FAA that support information exchanges and different protocols between operators and stake holders including FAA so that the Operators can work together cooperatively.

At the center of UTM architecture is the UAS Service Supplier (USS). USS is in charge of conduction an UAS operation while maintaining all the rules and regulation. USS is responsible for getting operations approved and executed in a timely manner.USS acts as the communication bridge between different service providers and UAS operations. It is envisioned there will be multiple USS operating in the same geographical location operating co-operatively (for eg. Amazon and Walmart both can act as USS and deliver products to the same neighbor hood).

Different USSs also communicates with each other relevant data needed for co-operative UAS operations.

UAS operations are confined inside "operational volumes". These are 4-D regions including airspace volumes and time. It is the responsibility of the UAS operator to safely conduct the UAS operation inside the approved volume. Any discrepancies are reported to the USS which handles each case. All flights in UTM TCL1 to TCL3 have had a single vehicle inside an approved operational volume.

Future operations will need many UAVs flying autonomously sharing approved volumes. In the current set up the operator is responsible for completing a particular operation within the stipulated time. As the density of operations increases, one operator will be in charge of multiple UAVs. These vehicles will operate autonomously BVLOS with minimal supervision.

It is assumed that BVLOS UAS share responsibility with other similar vehicles and manned aircraft for collision avoidance. Vehicles should have on-board collision avoidance algorithm to operate BVLOS in dense urban regions. One of the means of on-board collision avoidance is V2V communication between vehicles flying in close proximity.

Other requirements from FAA include that the operator should be able to track their vehicles in almost real time. Operators should notify FAA if the vehicles are in violation of the approved volume. V2V communication is an important aspect in case of declaration and receiving of emergency messages. This paper will describe such an operational architecture using V2V communication.

B. Contributions of this paper

Keeping the different needs and constraints of UTM in mind we have developed a V2V communication protocol that can handle cooperative UAS operation in UTM TCL-4. The NASA SAFE50[11] study aims to develop a complete point design that can faithfully adhere to the UTM rules and regulations and design a complete autonomous system that works in UTM TCL4. With in the SAFE50 architecture there is a requirement for real-time communication between different vehicles for on-board path planning. In this paper we have developed and tested a V2V communication architecture that is consistent with the UTM protocols and is useful for real-time path planning and handling other emergency vehicles.

The rest of the paper is arranged as follows: Section II describes the overall communication in UTM and why V2V communication is important. Section III describes the implementation details in the Reflection simulation architecture. Section IV describes the different path planning techniques we have utilized. Section V shows the results of different applications where V2V messages is used by the path planning algorithms. Finally section VI concludes the paper with some recommendations for future flight tests.

II. Communication in UTM TCL4

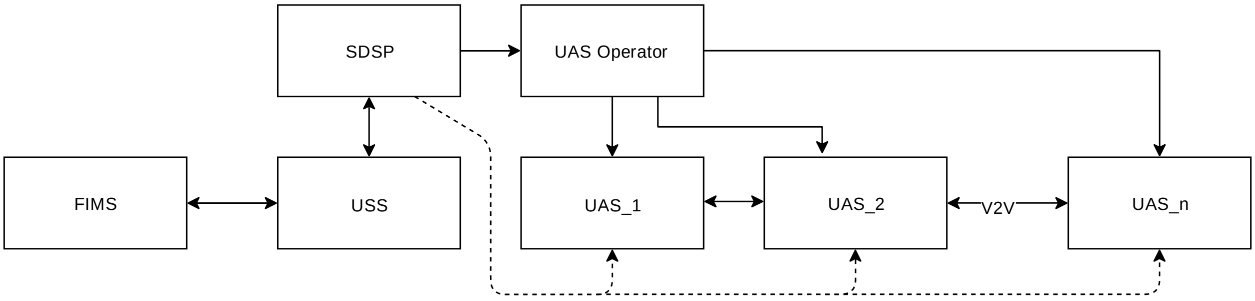


Figure 2. Proposed Communication Architecture for TCL 4.

The primary means of communication and coordination between Operators, FAA, and other stakeholders is through a distributed network of highly automated systems. Figure 2 shows the proposed communication architecture. The FAA makes real-time airspace constraints available to UAS operations via FIMS to the USS. The operator is responsible for managing their own operations safely within these constraints.

Supplemental Data Service Providers(SDSPs) give real time information like weather and traffic to the USS and UAS operator. It is still under consideration weather that data could be useful to the UAS itself. For real time path planner that information could be useful in terms of alternate route planning. In this particular paper we are not considering that SDSP has direct communication with UASs. Rathe we are only concerned with the vehicle to vehicle communication in this paper.

Operators have bidirectional communication with all the UASs. It helps to communicate plans and broadcast important message to the entire system. Also it is important to get telemetry information from individual UAS to make sure that the 4-D volume constraints are satisfied at all the times.

For autonomous operations BVLOS multiple vehicles will be flying in close proximity. It is imperative that the vehicles share information for safe and efficient flight in these operations among themselves.

A. Vehicle to Vehicle Communication

Vehicle to vehicle communication is important for vehicles flying in close proximity to avoid each other and to share emergency messages. V2V communication is being investigated for both manned and unmanned aerial vehicles. Many different options are being investigated in the industry like ADSB, wi-fi, DSRC etc.

In this paper we are implementing a TCP/IP based protocol which can be implemented with the current state of the art infrastructure. WE wanted to simulate delay and lag associated with the V2V communication and their effects in vehicle navigation. To see other potential ways to implement V2V communication see.[12,13]

We have used socket programming to implement the V2V communication in this paper. Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server. A server client setup is used to communicate different vehicle in the network. We have chosen such a framework as it is easy to implement, readily portable to hardware and easily expandable to other future hardware.

The next section describes our simulation infrastructure. We use the Reflection simulation architecture. The V2V communication module has been developed as a stand alone plugin for the Reflection architecture.

III. Implementation

The Reflection Architecture is a real time component based plug and play architecture for rapid development of embedded vehicle systems, developed here at NASA Ames.[14] This system has been used in many applications at NASA and we have used this system with UTM to build an end-to-end simulation of UTM TCL4 simulation.

Detailed design and implementation of a fully autonomous and programmable autopilot system for small scale autonomous unmanned aerial vehicle (UAV) aircraft using the Reflection architecture is describe here.[15] In this paper we discuss in details the subsystems that are relevant to both path planning and V2V communication.

Figure 3 shows the Reflection architecture for the overall system. With the Reflection plug-and-play architecture, simulation and hardware can be mixed on the fly for in-situ simulation testing of hardware components at any level of granularity. As shown in figure 3, the same architecture can be be converted into flight tests. The Flight Management System(FMS) handles the overall data management.

The main components of the Reflection software architecture can be divided into sensing and perception, decision making and planning and control. The sensing and perception unit consists of SLAM processing, Object detection and vehicle sensors. The planning and control subsystem comprises of path planning subsystem, local planner and autopilot. The Decision making module is responsible for the overall behavior of the system. The Flight Management System(FMS) handles the overall data management (figure 3).

The control system of the architecture is an instance of the Autopilot System (AP) class as shown in figure 3. The AutopilotSystem class is responsible for communicating with the rest of the Reflection system and maintaining the two main objects in the system: the FMS (flight management system) and the Controller. The FMS is responsible for maintaining the list of commands which specify FMS mode instructions. The mode instructions are used by the FMS to provide targets to the controller. The controller is responsible for implementing the control loops which control the aircraft through the vehicle?s actuators.

The Decision Making(DM) module is responsible for the overall behavior of the UAS. The DM communicates with the rest of the UTM system and ensures the overall feasibility if the system. From several feasible Trajectories (1,2,...n) the DM decided the final waypoints the vehicle has to fly. Together with the list of way-points and the approved volume around them, the Local Planner module generates feasible trajectories inside the approved volume avoiding other vehicles sharing the same volume.

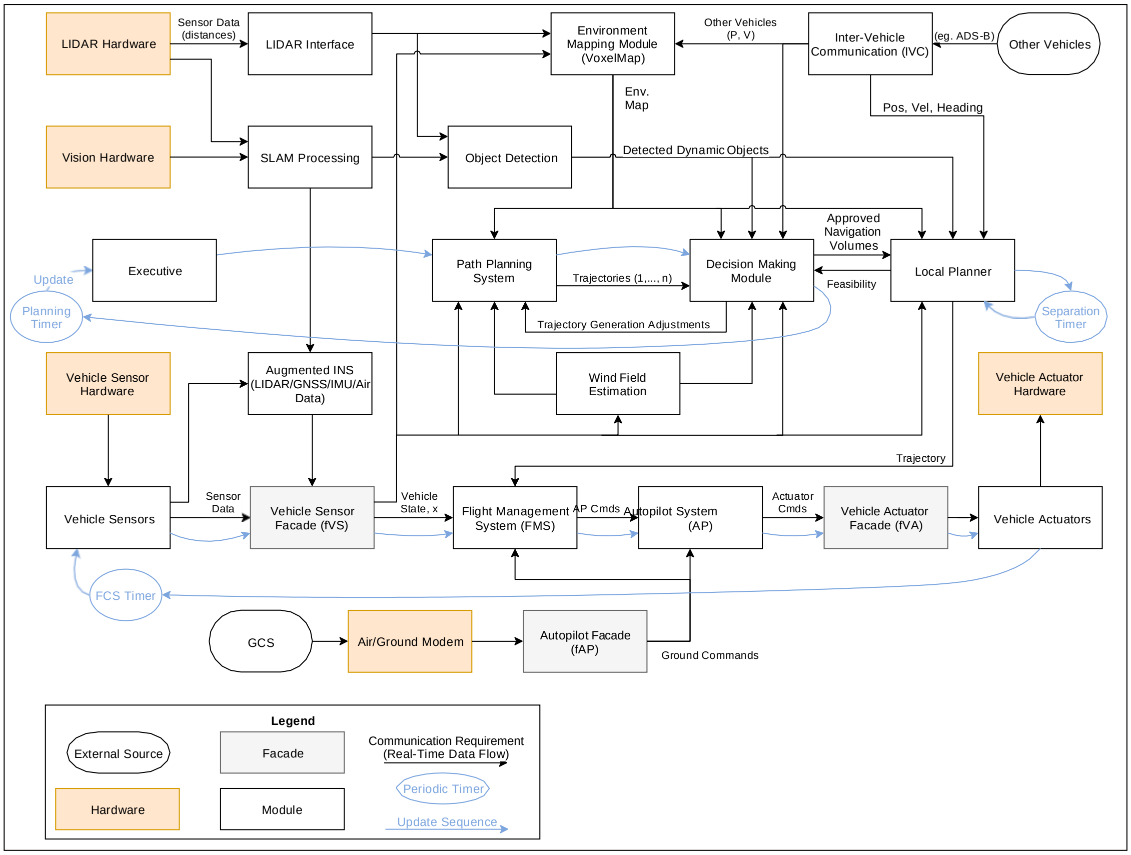


Figure 3. Reflection Software Architecture Diagram.

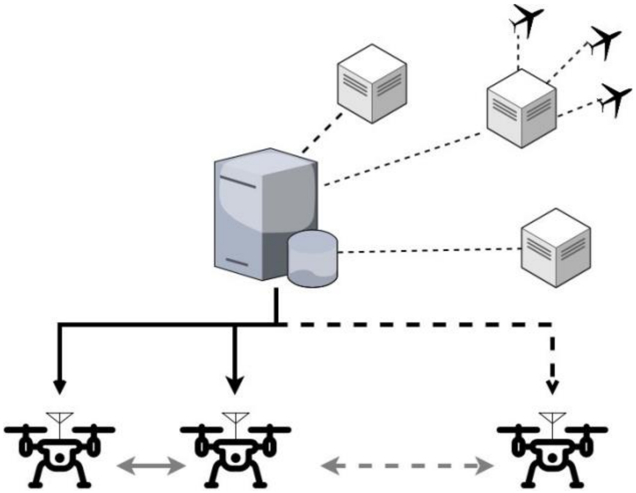


Figure 4. Inter Vehicle Communication.

A. V2V Subsystem

The V2V subsystem is written as a TCP/IP protocol using a socket interface. The V2V subsystem is designed as a server client interface. It is not necessary to have a sever/client setup for V2V communication. In fact most short range vehicle to vehicle communications are one on one connection. But we chose to simulate a V2V server as a server client setup as it gives us enough flexibility to simulate many of the desired properties of the vehicle to vehicle communication like delays in either client or servers or mixed packages.

Servers and clients have very different behaviors. A server starts as a separate node in the reflection architecture. The server runs on a separate thread and does not depend on the system updates. The sever initializes the socket, creates a socket and waits for clients to connect to the socket.

A client on the other hand creates a socket. It looks for the server to bind to. Once it has connected to the server it sends the messages through a TCP connection. Figure 4 shows a notional diagram of V2V communication. Figure 4 shows that the server is centrally located and has bi-directional communication with all the connected vehicles. Each vehicle participating in the flight registers as a new client with an unique id. Each vehicle has access to the messages send through the V2V server. Thus all vehicles are aware of the vehicles in its close vicinity.

A Supplemental Data Service Provider (SDSP) can provide such a service. Dedicated short-range communications(DSRC) communication protocols can also be used for vehicle to vehicle communication. We have designed our system to simulate the particular hardware which will be used in test flight. We can modify our simulation setup to make it behave as a DSRC connection.

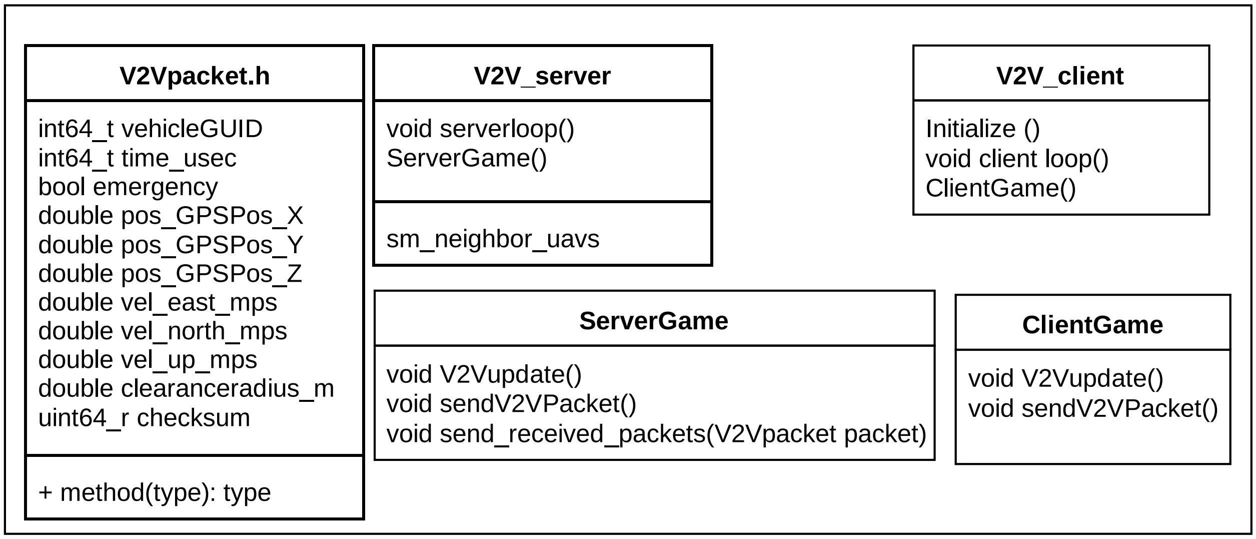


Figure 5. Reflection Software Architecture Diagram.

Figure 5 describes in details the classes and the V2V message packet. Each vehicle participating in the flight registers as a client. Each vehicle participating in a UTM flight is given a unique identification number. The V2V message packet contains the unique id of the vehicle, and its position and velocity in GPS co-ordinates. All the messages are time stamped and checksum is added to make sure we do not have dropped packages. Also as different vehicles will participate in UTM, each vehicle declares its own size. The variable *clearanceradius_m*, states the clearance distance needed for this vehicle for collision free trajectory. Every vehicle connecting to the server sends this information at a fixed rate. Before sending the data over the network the data is serialized and using standard TCP/IP protocol the data is send over the network.

The V2V server class runs in its separate thread. The server receives V2V packets from all the individual clients. Once it receives the packages it stores them in its buffer. During the V2VUpdate() in the ServerGame routine the server pushes the current list of messages to all the connected clients. In this way all the vehicles get information of all the other vehicles participating in UTM.

For a realistic set up of DSRC communication, it is possible to modify the send-to-all routine. It is possible to only send V2V information to vehicles in close proximity. This enables us to simulate DSRC communication.

Figure 6 shows the sequence diagram of the V2V communication module. The V2VServer::update() happens at 10 hz while the client update happens at 1 hz. The V2VServer maintains a list of all connected clients names

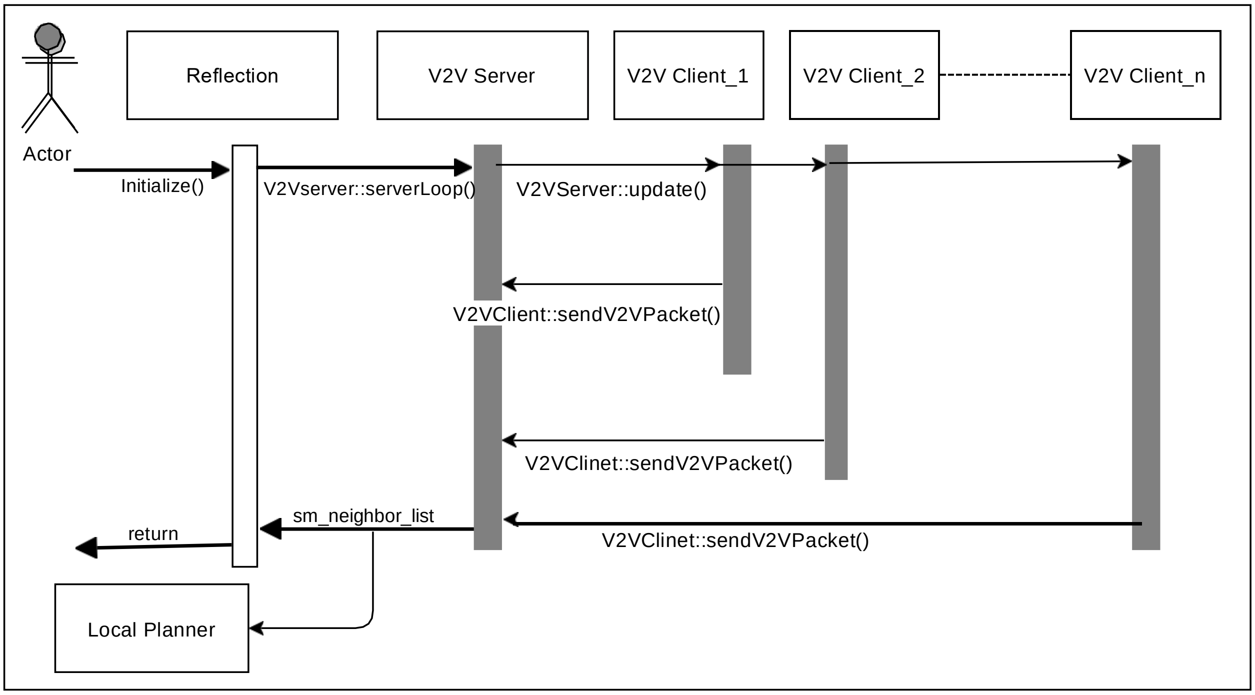


Figure 6. Reflection Software Architecture Diagram.

sm_neighbor_list. This is used by different modules in the Reflection architecture like Local Planner and Global planner.

1. *Vehicle Position Prediction*

The raw data obtained from the Vehicle to vehicle communication module is not useful for planning purposes. But since all the data is time stamped we can predict the future states of the vehicle based on its current position and velocity. A simple Kalman filter can be employed to predict the future states of the other UAVs if there are dropped packages. The covariance matrix will used as the uncertainty of position of the vehicle. This can be used as volume which is projected in the future that a particular vehicle is expected to occupy in the near future. Vehicles equipped with different path planning algorithms avoids this predicted volumes occupied by the vehicle.

In this paper we describe a avoidance volume based on perfect knowledge of the vehicle state. Section V-C shows the construction of avoidance volume based on perfect knowledge of vehicle state and velocity time stamped. This will be expanded in future research with dropped packages and different time delays.

IV. Path Planning

We have investigated different path planning algorithms for finding efficient collision free paths for vehicles communicating over V2V messages. Motion planning is an active area of research in the robotics community. The main challenge of this work is the integration with the larger UTM systems. In this section we present two different approaches that we have investigated.

A. Velocity Obstacle

Velocity Obstacle is a motion planning algorithm particularly designed for moving obstacles.[16] We implemented a collision avoidance module using Velocity Obstacle.

The collision avoidance module uses the current vehicle state (position, velocity, position errors) and V2V messages from nearby UAVs to calculate collision free velocities. The velocity obstacle concept for two UAVs is shown in figure , where V_A and V_B are the velocities of two UAVs, S_A and S_B are the desired separation distances and U_A, U_B are the position uncertainty of each UAV.

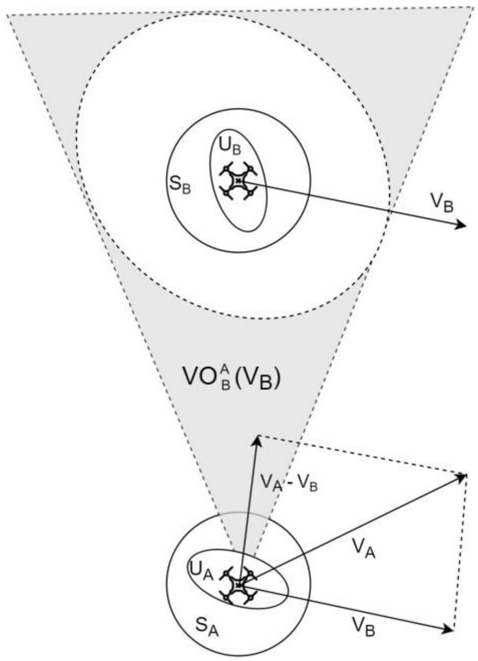


Figure 7. Velocity Obstacle.

The UAV calculates the desired velocity V_d for tracking the planned mission. This velocity V_d is used as an input to the collision avoidance module which outputs a collision free velocity Vc to the autopilot. If a collision is predicted, a new velocity vector V_c is calculated while trying to minimize the difference between desired velocity V_d and the newly computed velocity V_c. A Kalman Filter position predictor is used to predict the position and velocity of other UAVs when no new V2V messages are received or messages are lost. This approach can lower the V2V communication requirements and improve the collision avoidance when a large number of UAVs are in communication range.

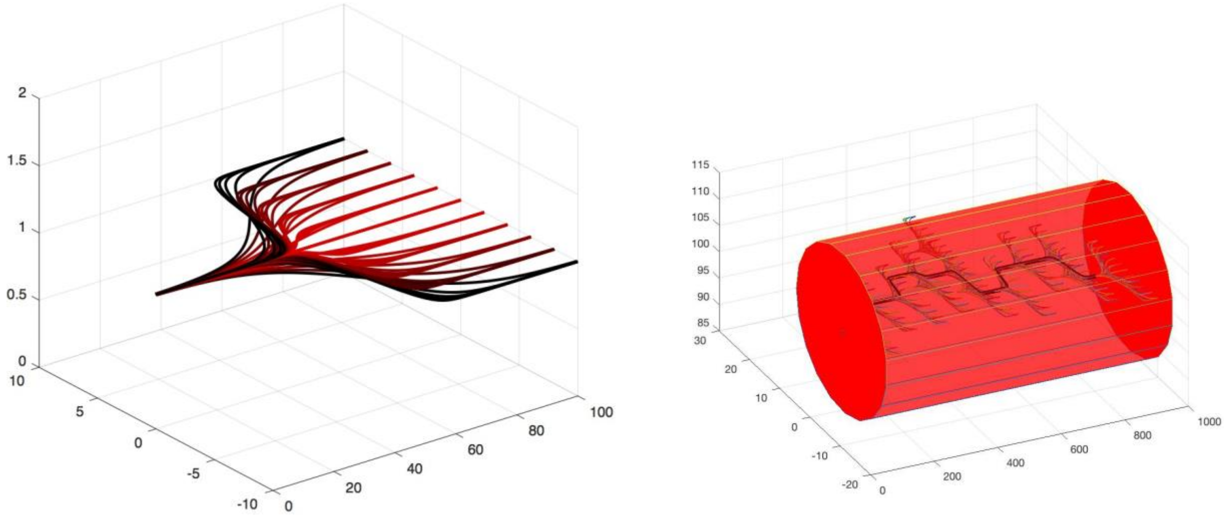
The Velocity Obstacle method was successful in finding collision free paths with other UAVs sharing the same air space. We also developed a tree based path planning algorithm which considered other UTM constraints. The next section briefly describes the Tree based planning algorithm.

B. Tree Based Path Planner

Tree based trajectory planners are very popular for mobile robot applications. In this methods a tree is grown from starting configurations using all the possible inputs. Each branch in the tree reflects the possible trajectories. There are many variations of tree based trajectory planners Rapidly exploring random trees (RRTs).[17] We use a tree based trajectory planner described in.[18]

In this method, a set of reachable states, called motion primitives, are calculated from the vehicles current states. Using the set of motion primitives, the tree is grown iteratively. Each node in the tree consists of an inertial position and a weight is assigned to each node. The weight is a combination of accumulated cost to reach the node and a heuristic cost to get to the goal. Other cost like, distance to other objects, cost to travel in different wind conditions,[19],[20] etc can also be added to the cost function. The tree algorithm picks the node that has the minimum cost and grows the tree from the chosen node. This process is continued until a goal is reached. This method has been proved to be resolution complete and optimal if the cost function is admissible.[18]

We show results of using both the methods of path planning where different vehicles communicate over a V2V server. The next section describes the problem and simulation results obtained from high fidelity Reflection software.



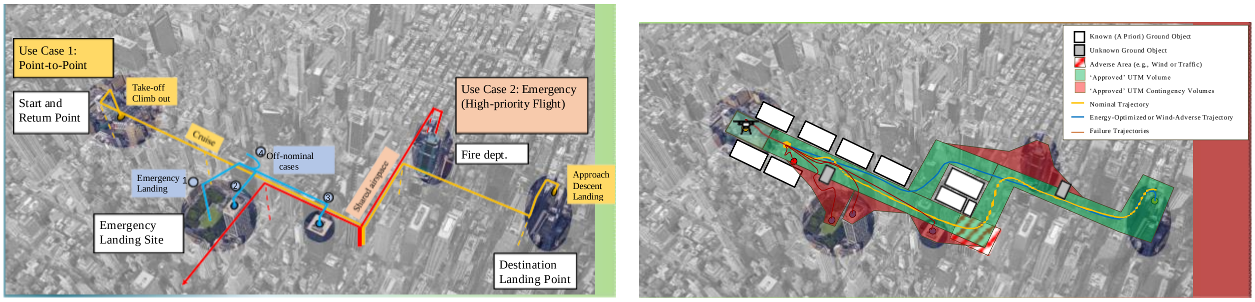
(a) Local Plan generated inside the approved volume

(b) Top view of the local plan that was selected.

Figure 8. Local Plan generated at each instant of time

V. Results

A. Use Case



(a) A point to point UAV operation example in dense urban operations.

(b) Approved volumes by UTM is designated in Green and shows that it is approved before the flight.

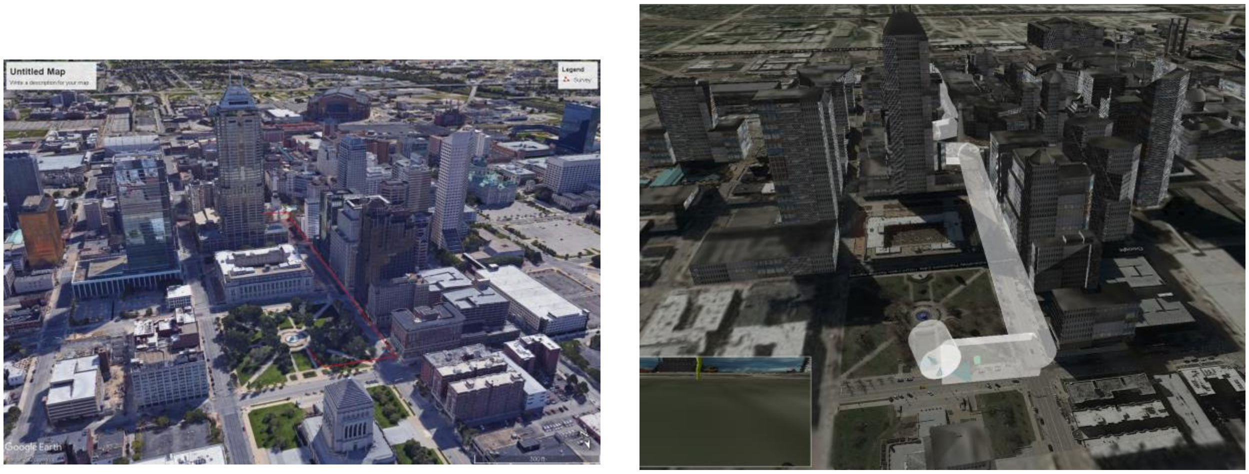
Figure 9. Typical UTM TCL-4 Operation Scenario

We simulated a point to point BVLOS UAS operation in a urban environment. Package delivery[1] is one of the motivating examples for UAS operations in dense urban environments which will require a point-to-point BVLOS operation. As shown in figure 9a, a typical package delivery flight will consist of take-off from designated launching site, travel to a designated delivery point and return back to the launching site. The vehicle have to encounter known/unknown objects along the way and other vehicles or emergency vehicles operating in the vicinity.

The vehicle will have to operate inside an UTM approved volume during the whole operation. As shown in 9b the typical TCL-4 operation will have known a-priori obstacles which the approved volume avoids. Also along the way there would be UTM approved contingency volumes which can be used only during emergency. This can include local parks or rooftops. See current UTM operations and concepts at,[21].[22]It is assumed that the vehicle can operate autonomously and safely within this approved volume.

Downtown Indianapolis in the Whole Sale District offers an example of typical urban neighborhood for UTM TCL-4 operations. Here we consider an autonomous flight between two points as shown in Figure 10. The waypoints are shown as *red* line in figure 10a. It is assumed that the UTM will approve a volume around the list of waypoints before the flight. Also there is a designated contingency volume that the vehicle must use in case of emergency situations. We consider the park shown in 10a, the region around the take-off location as the only contingency volume.

The Reflection simulation architecture simulates the downtown Indianapolis as shown in figure 10b. The urban



(a) List of waypoints in Downtown Indianapolis (Google Earth).

(b) Reflection software simulating the same environment with UTM approved volume.

Figure 10. Algorithm being tested in reflection architecture.

landscape rendering and collision model was created for large section of downtown Indianapolis, see[23] for details. Figure 10b shows the screen shot from Reflection software and the UTM approved volume for the given list of waypoints. The contingency volume is the park surrounding the take off sector.

All vehicles participating in the autonomous flight communicates through V2V messaging. The task of the local planner is to plan collision free paths inside approved volumes and safely navigate the vehicle from start to end.

It is assumed that the prior permission to fly was obtained and the Decision maker module has selected the current list of waypoints and this has been passed on to the Planning subsystem. Now the task of and autonomous UAV is to safely fly along the approved UTM volume while avoiding static and dynamic obstacles. If there is an emergency situation the vehicle should be able to travel to the contingency volume.

B. Static Obstacle avoidance using Velocity Obstacle

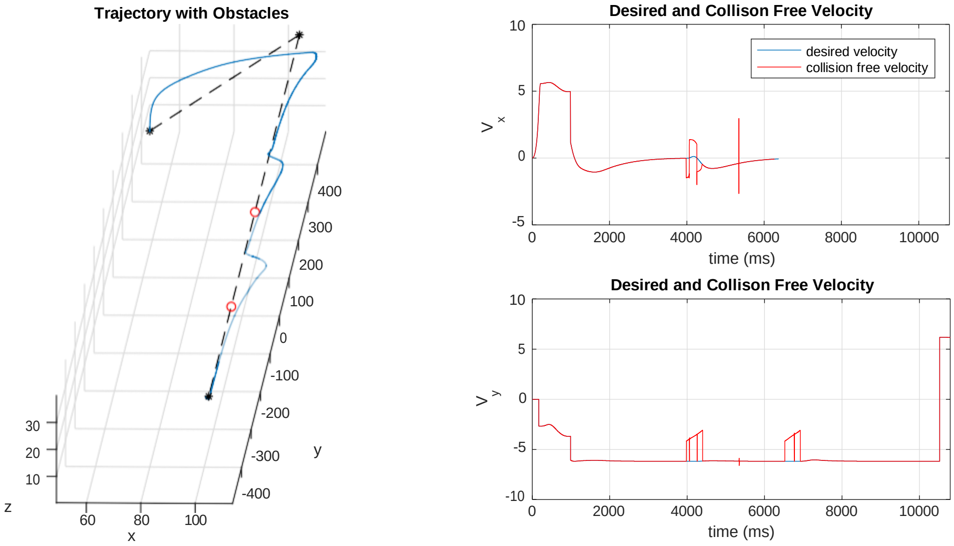


Figure 11. Velocity Obstacle.

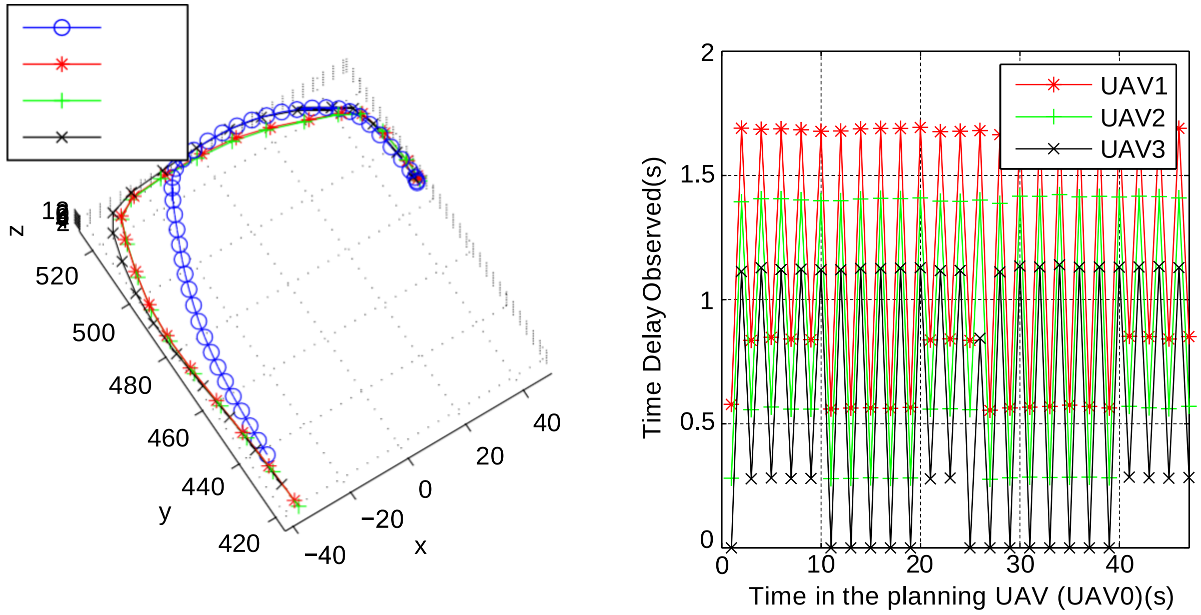
We show a simple example of Velocity obstacle being used as a local planner for Static Obstacle. The vehicle starts at local co-ordinates ($x = 100, y = 500, z = 0$) with a destination at ($x = 50, y = -500, z = 40$) with obstacles at ($x = 100, y = 100, z = 40$) and ($x = 100, y = 100, z = 40$)(all values in local co-ordinates in meters., figure11)

As shown in Figure 11 the Velocity Obstacle method is successful in finding collision free paths for for static obstacles in the Reflection architecture. The location of the static obstacles are simulated as static vehicles sending V2V messages.

Figure 11 shows that the velocity obstacle method has been successful in avoiding static obstacles. Figure 11b shows the collision free velocity output from the Velocity Obstacle algorithm. This simple example shows that the V2V messages can be used for obstacle avoidance and Velocity obstacle method can be reliably used as a path planning algorithm.

Other constrains from UTM are difficult to handle by VO method. We use tree based planning methods to handle moving obstacle.s The next section describes multiple vehicles flying in Reflection communicating over V2V messages.

C. Multiple Vehicle Simulation in Reflection



(a) Trajectory of other UAVs as observed by the planning UAV(UAV0)

(b) Time delay in receiving V2V message from other UAVs as observed by the planning UAV(UAV0) .

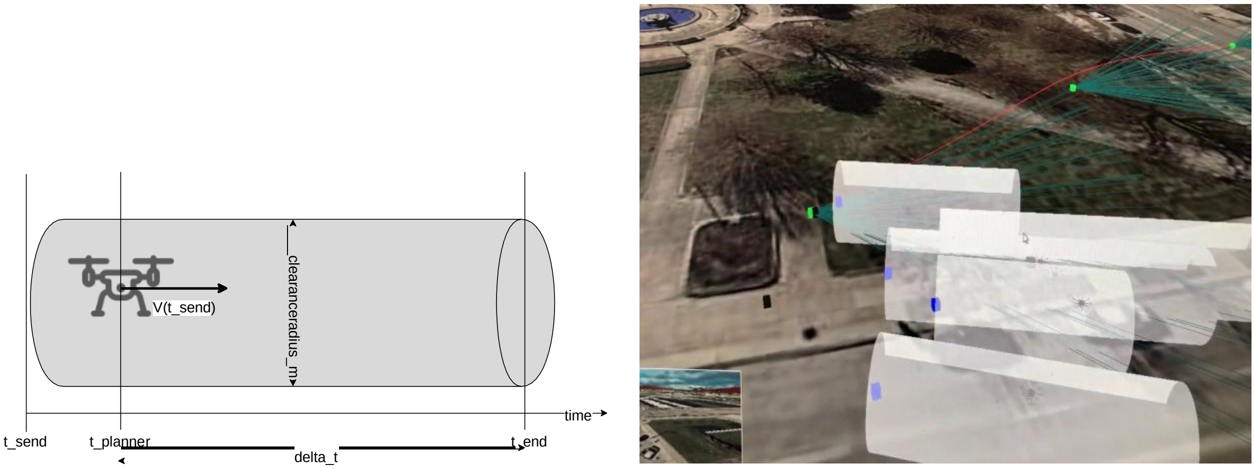
Figure 12. V2V communication received by the planning UAV

A multi-UAV simulation was designed in the Reflection architecture. As mentioned in the Section III-A the V2V communication class communicates with all the connected vehicles using a TCP/IP protocol. The vehicles do not communicate their entire trajectory or intent. Each V2V message contains position and velocity information time stamped for each vehicle. The message also contains desired clearance radius for each vehicle.

The planning vehicle (UAV0) maintains a list of all the connected vehicles. The positions that the planning UAV receives has to be interpolated for obstacle avoidance. Figure 12 shows a sample run with 4 UAVs (UAV0- UAV4) with UAV0 being the planning UAV. UAV0 receives communication over the V2V server about the other vehicles. Figure 12 b shows the delay in receiving the V2V message as observed by UAV0. The significant time delay observed makes it important to estimate the position of other vehicles from the planning UAV.

The local planner estimates the position of the all the other vehicles at the current time by interpolating the position with velocity information and adjusting the time delay. Also the local planner estimates the possible position of the vehicle during the next projected time interval. These two positions along with the clearance radius is used to define an avoidance volume for each of the connected vehicles (See figure 13a for explanation.).

Figure 13b shows the implementation in Reflection. The black dot in the figure is the position of each vehicle that was received by the vehicle at t_{send}. The blue dot represents the position interpolated at the time that corresponds to



(a) Avoidance Volume

(b) Avoidance volume implementation in Reflection

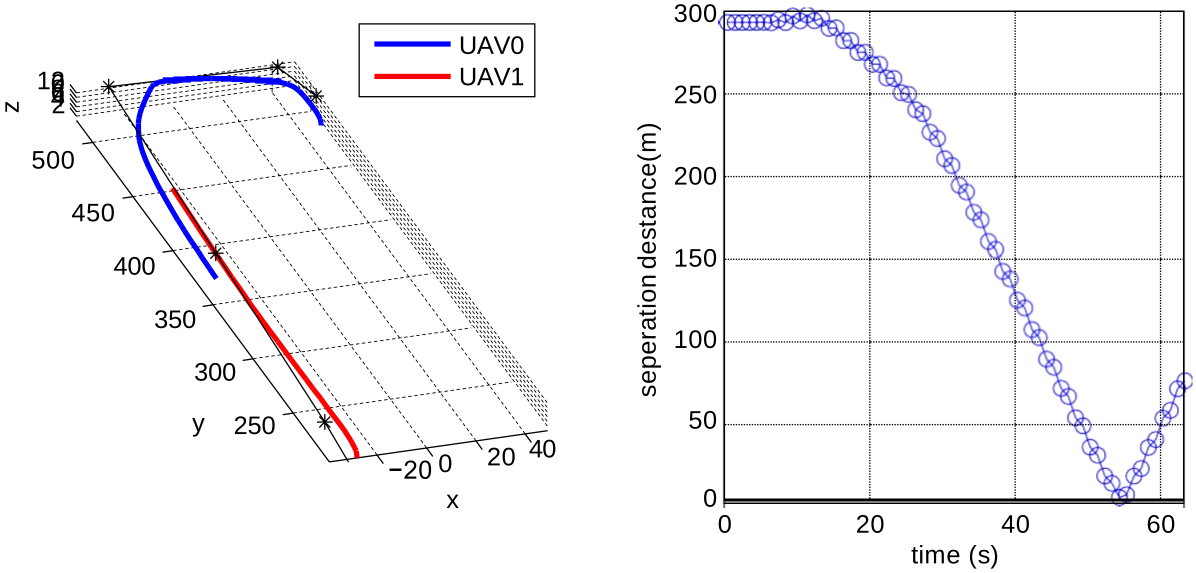
Figure 13. Estimated Volume to avoid for other UAVs communicating over V2V

the planner. And the white translucent volume is the avoidance volume for each vehicle. The recursive tree planner needs to avoid all the avoidance volumes while maintaining the UTM constrains.

1. *Avoiding Other Vehicles*

Consider the scenario SectionV-A. The waypoints are shown as *red* line in figure 10a. But in this case we consider flight by two UAVs starting at opposite ends of the the approved waypoints. It is assumed that the UTM will approve a volume around the list of waypoints before the flight.

Our planning UAV (UAV0) follow the same set of waypoints as described in sectionV-A. While the other UAV(UAV1) follows the reverse set of way-points. Both the UAVs share their position and velocity using the V2V communication described above.



(a) De-conflicted Trajectories of the two UAVs

(b) Separation distance between the two UAVs.

Figure 14. Implementation in reflection

Figure 14 shows the collision free trajectories of the two UAVs sharing the same approved UTM volume. The separation distance between the two UAVs are also plotted. The minimum separation distance between the UAVs maintains the clearance radius requested by UAV1 through the V2V messages.

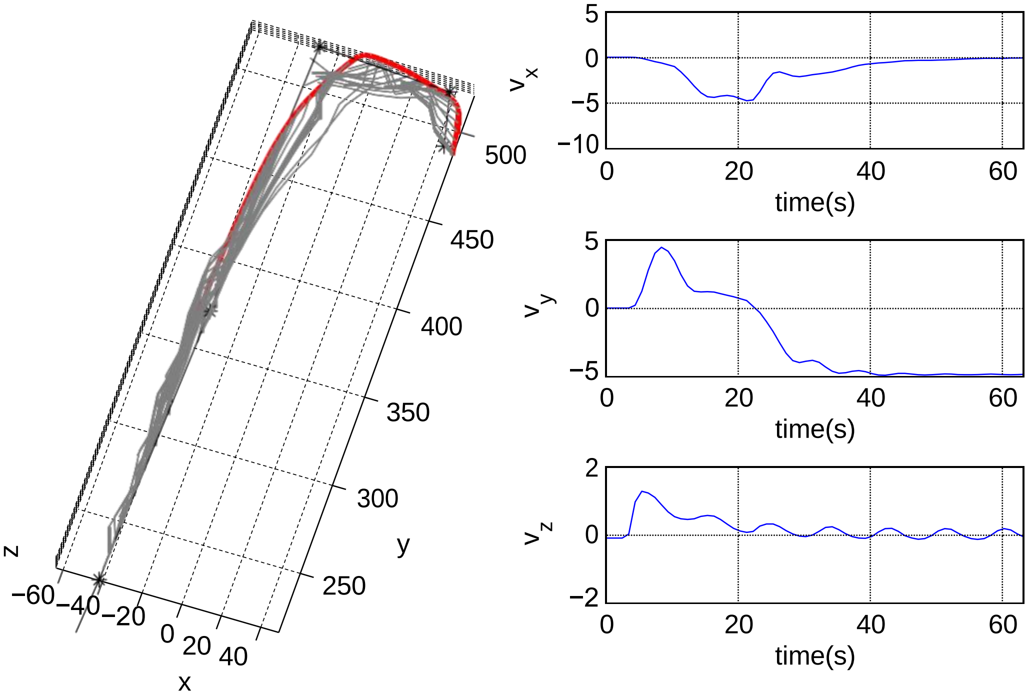


Figure 15. Trajectory and Velocity solution of UAV0 using the tree based trajectory planner

Figure 15 shows the trajectory and velocity of the planning UAV (UAV0). Figure 15 also shows the evolving trajectories of UAV0 with time. The recursive tree planner thus successfully finds de-conflicted trajectories with other UAVs sharing the same UTM approved volume. This technology enables vehicles operating in BVLOS UTM TCL-4 flights with on-board real time re-planning capabilities and sharing the same volume with other similar vehicles.

2. *Avoiding Emergency Vehicles*

As noted in Section III-A the V2V message contains information regarding emergency messages. This is particularly important for applications like law enforcement or medical emergency. These vehicles are given higher priority and all vehicles participating in UTM TCL4 should have contingency plan when emergency vehicles enter the approved volume.

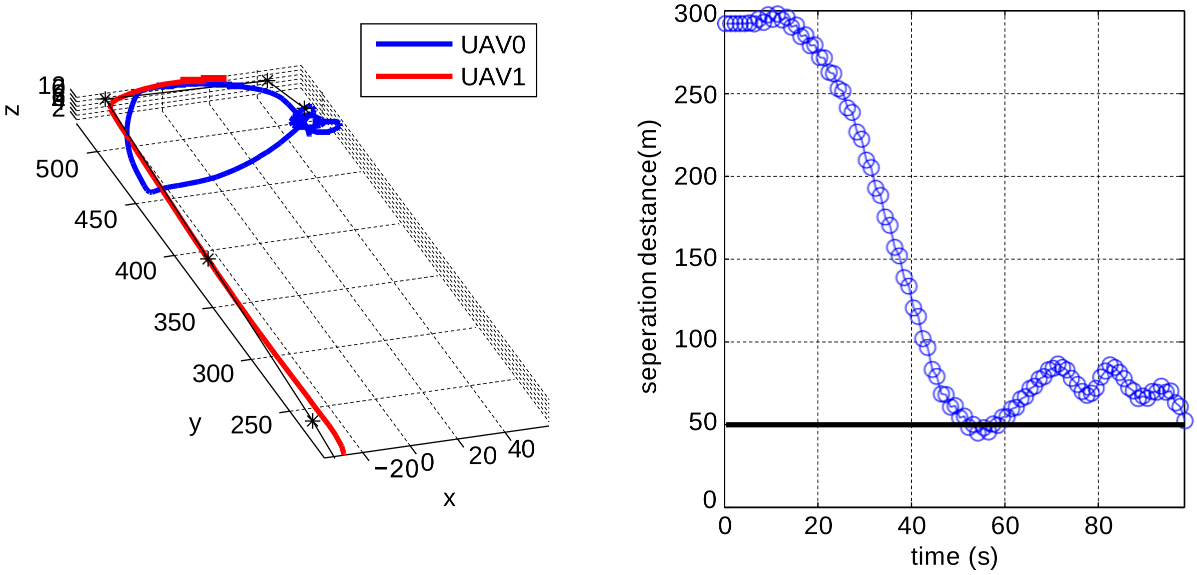
We again consider the same simulation setup as described in the previous scenario, except UAV1 is an emergency vehicle and broadcasts that information over V2V messages. The emergency vehicle also requests a clearance radius of 50 meters.

In this situation the planning UAV must use the contingency volume. So instead of planning collusion free paths around the other vehicle, as-soon as the planning vehicle comes in close vicinity of the the emergency vehicle, it returns back to the contingency volume and hovers near the take off position.

Figure 16a shows the trajectory of the two UAVs. The emergency vehicle(UAV1) continues on its trajectory while the planning vehicle choses to return to the contingency volume. Thus V2V messages can be successful used to plan to handle emergency situation in UTM TCL-4.

VI. Conclusion

As concepts of UTM is evolving around flying UAS in dense urban regions, vehicle to vehicle communication has emerged as a possible solution to handle different problems arising from urban operations. This paper has demonstrated an end to end simulation for urban UAS operations with vehicles communicating over V2V server. We have



(a) Trajectories of the emergency vehicle and the planning UAV

(b) Separation distance between the two UAVs.

Figure 16. Trajectory when an emergency vehicle enters the approved volume

demonstrated on-board path planning using V2V communicating and on-board path re-planning to handle emergency vehicles. The paper described in details a V2V server-client design using TCP/IP protocol. The entire system is tested in high fidelity Reflection simulation architecture.

Future research will focus on managing large fleet of vehicles using V2V. Also the effects of dropped V2V messages on path planning and decision making will be explored. Various estimation schemes will be explored which can be used on-board and real-time. Flight tests will be conducted to verify this set up can be used in flights for TCL4.

Acknowledgments

The authors would like to thank our colleagues in the NASA SAFE50 and UAS Traffic Management (UTM) projects for their collaboration and insight.

References

[1] G. Hoareau, J. J. Liebenberg, J. G. Musial, and T. R. Whitman, “Package transport by unmanned aerial vehicles,” Aug. 15 2017, uS Patent 9,731,821.

[2] J. Katrasnik, F. Pernus, and B. Likar, “A survey of mobile robots for distribution power line inspection,” *IEEE Transactions on Power Delivery*, vol. 25, no. 1, pp. 485–493, 2010.

[3] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, “Control of multiple uavs for persistent surveillance: algorithm and flight test results,” *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1236–1251, 2012.

[4] S. M. Adams and C. J. Friedland, “A survey of unmanned aerial vehicle (uav) usage for imagery collection in disaster research and management,” in *9th International Workshop on Remote Sensing for Disaster Response*, 2011, p. 8.

[5] J. T. K. Ping, A. E. Ling, T. J. Quan, and C. Y. Dat, “Generic unmanned aerial vehicle (uav) for civilian application-a feasibility assessment and market survey on civilian application for aerial imaging,” in *Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2012 IEEE Conference on*. IEEE, 2012, pp. 289–294.

[6] F. A. Forecast, “Fiscal years 2011–2031,” *FAA, US Department of Transportation*, 2011.

[7] P. H. Kopardekar, “Unmanned aerial system (uas) traffic management (utm): Enabling low-altitude airspace and uas operations,” 2014.

[8] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. Robinson, “Unmanned aircraft system traffic management (utm) concept of operations,” in *AIAA Aviation Forum*, 2016.

[9] M. Johnson, J. Jung, J. Rios, J. Mercer, J. Homola, T. Prevot, D. Mulfinger, and P. Kopardekar, “Flight test evaluation of an unmanned aircraft system traffic management (utm) concept for multiple beyond-visual-line-of-sight operations,” 2017.

[10] “Concept of operations v1.0.” [Online]. Available: https://utm.arc.nasa.gov/docs/2018-UTM-ConOps-v1.0.pdf

[11] K. S. Krishnakumar, P. H. Kopardekar, C. A. Ippolito, J. Melton, V. Stepanyan, S. Sankararaman, and B. Nikaido, “Safe autonomous flight

environment (safe50) for the notional last 50 ft of operation of 55 lb class of uas," in *AIAA Information Systems-AIAA Infotech@ Aerospace*, 2017, p. 0445.

[12] M. L. Sichitiu and M. Kihl, "Inter-vehicle communication systems: a survey," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 2, 2008.

[13] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, "A survey of inter-vehicle communication protocols and their applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, 2009.

[14] C. Ippolito, G. Pisanich, and K. Al-Ali, "Component-based plug-and-play methodologies for rapid embedded technology development," in *Infotech@ Aerospace*, 2005, p. 7122.

[15] C. Ippolito, G. J. Pai, and E. W. Denney, "An autonomous autopilot control system design for small-scale uavs," 2012.

[16] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[17] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 995–1001.

[18] A. Chakrabarty and J. Langelaan, "Uav flight path planning in time varying complex wind-fields," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 2568–2574.

[19] A. Chakrabarty and J. W. Langelaan, "Flight path planning for uav atmospheric energy harvesting using heuristic search," in *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 8033.

[20] A. Chakrabarty and J. Langelaan, "Energy-based long-range path planning for soaring-capable unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1002–1015, 2011.

[21] M. A. Johnson, "Unmanned aircraft system traffic management (utm): Defining the future of the drone industry," 2017.

[22] J. Rios and M. Johnson, "Unmanned aircraft systems traffic management (utm) concepts and architecture overview," 2018.

[23] C. A. Ippolito, S. Hening, S. Sankararaman, and V. Stepanyan, "A modeling, simulation and control framework for small unmanned multicopter platforms in urban environments," in *2018 AIAA Modeling and Simulation Technologies Conference*, 2018, p. 1915.